# Extracting hidden anomalies using Sketch and non Gaussian Multiresolution Statistical Detection Procedures*

G. Dewaele[(1)], K. Fukuda[(2)], P. Borgnat[(1)], P. Abry[(1)], K. Cho[(3)]
[(1)]Physics Lab., ENS Lyon - CNRS, France [(2)]National Institute of Informatics, Japan [(3)]Internet Initiative Japan

## Abstract

This paper proposes a new profile-based anomaly detection and characterization procedure in order to promptly detect both short-lived and long-lasting low-intensive anomalies without any prior knowledge about the target traffic. The key feature of the algorithm is the joint use of a random projection technique (sketches) and multiresolution non Gaussian marginal distribution modeling. The former reduces the dimensionality of the data and is used to obtain the reference (i.e., normal) behavior among the sketches. The latter extracts anomalies in multiple time-scales. This detection procedure is used to blindly analyze a large-scale packet trace database collected on a trans-Pacific transit link from 2001 to 2006. Our tools can detect and identify a large number of known and unknown anomalies and attacks, whose intensities are very low (down to below one percent). The sketch procedure also allows real-time identification of the IP source or destination addresses in the detected anomaly, hence their mitigation.
**Key Words:** Anomaly detection, Traffic measurement, Sketch, Gamma multiresolution modeling

## 1 Introduction

A hot topic for Internet security lies in detecting attacks promptly and accurately and in defining mitigation policies. Often, anomaly detection and characterization is an involved task for several reasons. One needs to detect attacks when they are still at very low intensities, hidden amongst large volumes of regular traffic. Anomalies to be detected show an extreme diversity in nature (DDoS, flooding, flash crowds, worm outbreaks), in duration (from very bursty or short lived ones to long lasting ones), in targets and goals. Moreover, new varieties of anomalies are constantly appearing every day. Therefore, designing filters (based on a characteristic time scale for instance) that match a given known anomaly quickly turns obsolete. Last but not least, regular traffic in itself exhibits a wild variability (heavy tails and long range dependence) [14], which significantly impairs the detection of anomalies.

In the case (of interest in the present work) of a single-point measurement performed over a transit or backbone link, extra complications can be listed. Traffic on such links is likely to be strongly asymmetric because of the multi-homed nature of the network. This forbids the use of techniques relying on the observation of bidirectional patterns (SYN, SYN/ACK,...). Also, tools making use of joint spatial network-wide information (Origin-Destination patterns) are excluded [11, 12, 17]. Backbone links aggregate at very high levels a large variety of traffic of different natures, with new types of regular applications constantly appearing, yielding extra regular variabilities. This implies the increase of the likeliness of simultaneous occurrences of anomalies, of undergoing known major anomalies such as worm outbreaks and intensive large scale attacks, and of observing known and unknown anomalies (i.e., zero-day attack). Facing such a diversity requires the use of as little as possible prior information, regarding the nature of the anomalies. Moreover, the huge volume of traffic precludes to store long (or even short) term traces, and calls for the use of low computational cost, possibly online, real-time and on-the-fly techniques.

Anomaly detection methods are broadly classified into two complementary categories: signature-based vs. profile-based detections. The anomaly detection procedure proposed here belongs to the second class. It is based on combining two key ingredients: Sketches and non Gaussian multiresolution (or multiscale) statistical modeling. A random projection based on a hashing key of IP source or IP destination divides a set of traffic data into sketches (sub-groups) to search for deviation in a statistical multiresolution modeling amongst the collections of sketches. Non Gaussian multiresolution statistical modeling is used to extract the shaper parameter of the correlation distribution of each sketch for multiple time-scales, and computes the self-reference of the entire sketch set. When the Mahalanobis distance of a sketch from the reference exceeds a threshold, the sketch is detected as an anomaly. Then, the corresponding IP

*Work supported by Strategic International Cooperative Program between CNRS and JST, and by the French METROSEC research project.

addresses in the detected anomaly are identified by the reverse hash function of the sketch.

The proposed method does not require *a priori* knowledge about the traffic because it compares the behaviors of sketches in the same data set. Our method works independent of the volume or time-scale of an anomaly, and thus, is able to detect low-intensive and/or long-lasting anomalies, even in one-way highly aggregated traffic. Our algorithm uses only IP addresses and rough packet arrival times so that no deep packet inspection is necessary, and is designed for near real-time processing at a single measurement point in a backbone.

This detection procedure is used to blindly explore the large scale MAWI packet trace database [6] in order to label the traces with anomalies to promote further research on them. This database stores daily packet traces from a trans-Pacific transit link over more than 6 years. The task of anomaly identification and labeling had not been previously done on this database, except for only rare prominent or huge anomalies. Results reported in Section 4 show that our procedure enables to detect and characterize anomalies, found to be in (unexpectedly) large number and variety. Also, our tool performs meaningful detections despite the fact that most anomalies have low intensities (below 1% in packet number), are hidden amongst many aggregated flows and often occur simultaneously. Identification of the targeted or faulty IP addresses is also fully operational so that prompt reaction and mitigation is possible. The proposed detection relies on little prior (network or statistical) information, hence is robust against significant traffic evolution along the years. Moreover, it enables not only the detection of known anomalies but also the discovery of a number of unknown and unexpected traffic flows, be their nature legitimate or not remains an open issue.

## 2 Related work

Profile based statistical anomaly detection is an active field of research [2, 5, 7]. To keep the amount of data manageable, a number of contributions make use of only high-level traffic measurements, e.g., SNMP data [2], or IP Flow data (often sampled NetFlow or FlowScan router reports) [2, 11]. This may imply losses of potentially important information. Hence, packet level information (IP packet (or volume) count process [7], or IP and/or TCP header [9]) can be preferred, enabling short reaction time (down to the packet arrival time, compared to a 5 min based flow report) and avoiding the recourse to potentially ineffective information (as with SNMP-MIB).

To deal with large amounts of data, anomaly detection procedures are nowadays more and more often relying on dimensionality reduction tools. In the context of network-wide measurements, Principal Components Analysis (PCA) [11], non linear manifold learning [15] are often used. This assumes that measurements over many points of the network are jointly available together with a centralized collection of data (as in [15]). This can be inefficient to react against sudden and localized attacks. Moreover, in the present work, we are interested in detection performed from the monitoring of a single transit or backbone link. Inspired by research on data streaming, the use of random projections or sketches, has been put forward in [10, 13] for change detection, information condensation or heavy hitter identification (see also [12]). In the present contribution, elaborating on [1], we make use of sketch procedures to split data into sub-traces and search for deviations in a statistical multiresolution modeling amongst the collection of sketches. Another major benefit of the use of sketch procedures lies in the possibility of identifying the attributes (e.g., IPsrc or IPdst) associated to the detected anomalies by the reverse hash function [13].

A central issue in profile-based detection lies in performing a proper statistical characterization of anomalies, as advocated in [2], so that detection necessarily consists of two steps: Model or predict an average reference traffic (with its naturally wild variability); Apply a decision rule for observing if analyzed traffic departs from the reference. Often, reference traffic is obtained as a prediction from past observations, (using, for instance, Holt-Winters forecasting [5, 10, 19], or Kalman filtering [17]) or direct observation [7]. Also, reference may be obtained from multi-link measurements when available [11, 17]. Framed in this methodological scheme, our contribution is original insofar as the reference traffic is determined from a single link measurement as an average over sketched time series, hence used as independent surrogate data. This avoids the recourse to traffic prediction, a highly difficult task because of its natural variability and long range correlations (see for instance [14]).

Single-link measurement profile-based detections have often been based on a specific statistical characteristic of the traffic such as spectral density or covariance [8], wavelet coefficients [2, 7], temporal features extracted from PCA, to list but a few (cf. [1, 16] and references therein). Also, the use of non Gaussian statistics has been used to seize the characteristics of normal traffic [14]. Recently, we went one step further promoting the use of non Gaussian statistics jointly over a large range of aggregation levels [16]. This multiresolution non Gaussian modeling is specifically tailored to design an anomaly detection procedure. Another originality consists of the fact that anomalies are not defined *a priori*, neither from a network mechanism or from a matched statistical pattern. An anomaly is here defined

as a statistical change in the correlation structure of the traffic in one sketch compared to that of other sketches. This original use of the sketch method enables the discovery of anomalies that are not known beforehand, and may never have been observed previously.

# 3 Anomaly detection method

The proposed anomaly detection procedure consists of the following steps.

**Step 1: Random projections (or sketches).** Packets are analyzed within sliding time-windows of duration $T$. For each time-window, let $\{t_i, \{x_{i,l}, l = 1, ...4\}\}$ denote the usual 5−tuple (arrival time stamp, IPsrc, IPdsst, Psrc, Pdst) for each packet $i = 1, ..., I$. Let $h_n$, $n \in \{1, ..., N\}$ denote $N$ independent $k-$universal hash functions, generated from different random seeds. Such $h_n$ is constructed using the fast-tabulation method presented in [18]. Let $M$ stands for the (identical) size of the hash tables. Let $A_i$ denotes the hashing key (in the present study, $A_i = \text{IPdst}_i$ or $A_i = \text{IPsrc}_i$). For each $h_n$, the original trace $\{t_i, \{x_{i,l}, l = 1, ...4\}, i = 1, ..., I\}$ is split into $M$ sub-traces, $\{t_i, m_{n,i} = h_n(A_i) = m, i = 1, ..., I\}_{n,m}$.

**Step 2: Multiresolution Aggregation.** The sub-traces $\{t_i, m_{n,i} = m, i = 1, ..., I\}_{n,m}$ are aggregated jointly over a collection of levels $\Delta_j, j = 1, ..., J$ to form the $X^{n,m}_{\Delta_j}(t)$ time series.

**Step 3: Non Gaussian modeling.** In a former work, we proposed [1, 4, 16] that the marginal distributions $f_\Delta(x)$ of aggregated traffic time series can be satisfactorily described using Gamma laws $\Gamma_{\alpha_\Delta,\beta_\Delta}$, i.e., non Gaussian distributions for positive random variables, defined as $\Gamma_{\alpha,\beta}(x) = \frac{1}{\beta\Gamma(\alpha)}\left(\frac{x}{\beta}\right)^{\alpha-1}\exp\left(-\frac{x}{\beta}\right)$, where $\Gamma(\cdot)$ is the usual Gamma-Euler function. The scale parameter $\beta$ mostly acts as a multiplicative factor (if $X$ is $\Gamma_{\alpha,\beta}$, then $\lambda X$ is simply $\Gamma_{\alpha,\lambda\beta}$). The shape parameter $\alpha$ controls the evolution of $\Gamma_{\alpha,\beta}$ from a highly asymmetric stretched exponential shape ($\alpha \to 0$) to a Gaussian shape ($\alpha \to +\infty$). More precisely, $1/\alpha$ can be read as a measure of the departure of $\Gamma_{\alpha,\lambda\beta}$ from the normal distribution $\mathcal{N}(\alpha\beta, \alpha\beta^2)$. Furthermore, $\Gamma_{\alpha,\beta}$ distributions are *stable under addition*: Let $X$ and $X'$ denote two independent $\Gamma_{\alpha,\beta}$ and $\Gamma_{\alpha',\beta}$ RVs, then $X + X'$ is $\Gamma_{\alpha+\alpha',\beta}$. This is of particular interest when related to the aggregation procedure: $X_{2\Delta}(t) = X_\Delta(t) + X_\Delta(t+\Delta)$. Indeed, should $X_\Delta$ be well modeled with a $\Gamma_{\alpha_\Delta,\beta_\Delta}$, then, it is expected that $X_{2\Delta}$ could be well modeled with $\Gamma_{\alpha_{2\Delta},\beta_{2\Delta}}$. Independence between $X_\Delta(t)$ and $X_\Delta(t+\Delta)$ would imply $\alpha_\Delta = \alpha_0\Delta$ and $\beta_\Delta = \beta_0$. Due to the correlations that exist amongst $X_{\Delta_j}(t)$ and $X_{\Delta_j}(t+\Delta_j)$, departures of $\alpha_\Delta$ and $\beta_\Delta$ from $\alpha_\Delta = \alpha_0\Delta$ and $\beta_\Delta = \beta_0$ are ascertained. Therefore, the Gamma description combined at various resolutions accounts not only for the marginal distributions of the aggregated traffic but also for its short-time statistical dependencies along time.
From the $X^{n,m}_{\Delta_j}(t)$, the corresponding collection of $\{(\alpha^{n,m}_{\Delta_j}, \beta^{n,m}_{\Delta_j}), j = 1, ..., J\}$ parameters are estimated (estimations being performed by means of standard sample moment procedures).

**Step 4: Reference.** For each $h_n$, average behaviors and typical variabilities are estimated as: $\alpha^{m,R}_{\Delta_j} = \langle\alpha^{n,m}_{\Delta_j}\rangle_m$ and $\sigma^2_{m,\alpha,\Delta_j} = \langle\langle\alpha^{n,m}_{\Delta_j}\rangle\rangle_m$, where $\langle\cdot\rangle_m$ and $\langle\langle\cdot\rangle\rangle_m$ denote the standard sample mean and variance estimators, respectively, computed from $m = 1, \ldots, M$.

**Step 5: Statistical distances.** Anomalous behaviors of the $\{(\alpha^{n,m}_{\Delta_j}, \beta^{n,m}_{\Delta_j}), j = 1, ..., J\}$ with respect to $\Delta_j$, are measured by computation of statistical distances from the reference behavior $\alpha^{m,R}_{\Delta_j}$. Many different statistical distances can be used, cf. [3] for a review. Here, we use the Mahalanobis distances (MD) to give the same weight to all scales:

$$D^2_{\alpha^{n,m}} = \frac{1}{J}\sum_{j=1}^{J}\frac{\left(\alpha^{n,m}_{\Delta_j} - \alpha^{m,R}_{\Delta_j}\right)^2}{\sigma^2_{m,\alpha,\Delta_j}}. \quad (1)$$

When $D_{\alpha^{n,m}} \leq \lambda$, $X^{n,m}_{\Delta_j}(t)$ consists of normal traffic; when $D_{\alpha^{n,m}} > \lambda$, $X^{n,m}_{\Delta_j}(t)$ is said to contain one (or more) anomaly(ies), where $\lambda$ is the detection threshold to be chosen. Let us put the emphasis of the fact that the use of a multiresolution distance implies that the detection procedure is not based on a change in volume of the traffic but rather on a change in its short-time correlation structure. Identical procedures are obtained for $\beta$, mutatis mutandis. However, detection based on this distance is not used in this paper.

**Step 6: Anomaly Identification by Sketch Combination.** To finish with, reversing the hashing procedures enables to identify the hashing keys associated to the detected anomaly(ies). When detections are performed in the $m$-th output of the $n$-th hash function, the corresponding attributes $A_i$ are registered in a detection list $A^n_i$. Combining the $N$ functions $h_n$ and taking the intersection of the $A^n_i$ yields a final list of attributes $A^o_i$ that correspond to detected anomaly(ies). In this respect, the use of $k$-universal hash functions, with $k \geq 2$, plays a key role as it guarantees that the average number of collisions between attributes $A_i$ diminishes exponentially fast with $N$, a collision consisting of any given pair of $A_i$ chosen at random amongst all possible *falls* within the same outputs $m_n$ for each of the $N$ $h_n$. If $A_i = \text{IPdst}_i$ and $N_{IP}$ is the total number of IP addresses observed in the analyzed traffic, the average

number of collisions reads: $\#_C = N_{IP}M^{-2N}$. Moreover, choosing $k \geq 4$ ensures that the variance of this $\#_C$ remains also small. Obviously, to lower the probability of random collisions and hence ensure relevant detections, we need to have this $\#_C \ll 1$. In Section 4, we observe that using only $N = 8$ hash functions is enough to ensure a correct identification of IPdst or IPsrc. Therefore, the procedure proposed here not only performs the detection of an anomaly (and provide the time windows in which it occurs), but also enables the identification of its $A_i$ attribute. This is a key feature allowing for the identification and classification, hence mitigation, of anomalies.

### 3.1 Performance and Validation

The validation and the assessment of the performance of the detection procedure described above raise two issues of different natures.

First, the detection tools are applied to a known database. Generally, statistical detection procedures always face the false positive/false negative trade-off, which, in the present context, is controlled by the choice of $\lambda$. Decreasing $\lambda$ amounts to allow detection when distances are smaller: This results in a increase of the correct detection rate, at the price of an increase of false negative, and vice versa. To assess such detection performance in terms of false positive/false negative scores, also referred to as receiver operational curves (ROC), we need to have a validation database, with known anomalies occurring at known times. Therefore, we created our own database consisting of actual traffic traces containing real anomalies [4]. They were generated by ourselves, in a controlled and reproducible manner, using real network tools such as trinoo, tfn2k,... and mostly consisted of mixed flooding DDoS attacks. These experimental set-up and database allowed us to show that the proposed tool exhibits very satisfactory ROC, even for anomalies whose volume is in the order of percent of the total traffic volume. The reader is referred to previous works [1, 4] for details.

Second, when applying detection tools to a database for which anomalies are unknown, one cannot compute false positive/false negative scores. Therefore, validation requires an *a posteriori* manual inspection of traffic. This is detailed in Section 4.

## 4 MAWI database Anomalies

### 4.1 MAWI database

The MAWI traffic repository of the WIDE project has been archiving raw packet traces measured over six years (from 2001-2006) at one of the trans-Pacific links (samplepoint-B, 18Mbps CAR) between Japan and the United States [6]. It is an academic network and the traffic on this link is mostly international commodity traffic to and from several Japanese universities. The database consists of 15-minutes-long traffic traces captured at 2pm, Japan time, by tcpdump and an IP-address anonymization tool. The traces amount to more than 2,000 traces or 600GB in size.

Table 1: Parameter description

| symbol | description |
|---|---|
| $n \in N$ | sketch number (number of hash functions) |
| $m \in M$ | sketch output number (size of hash table) |
| $\Delta_j$ | aggregation time scale $j \in J$ |
| $X_{\Delta_j}^{n,m}(t)$ | aggregated hashed time series for scale $\Delta_j$ |
| $\alpha_{\Delta_j}^{n,m}$ | the estimated $\alpha$ of Gamma function fit for $X_{\Delta_j}^{n,m}(t)$ |
| $D_{\alpha^{n,m}}$ | Mahalanobis distance of the estimated $\alpha$ |
| $\lambda$ | threshold value |

Analyses indicate that, for most of the days, the traffic on this link had been almost saturated. Also, they show that traffic is asymmetric in some periods, mostly because of route changes, and that some include major virus outbreaks. This study constitutes a first step towards labeling of anomalies contained in this database, and has explored data from weekdays, one day per two weeks from 2001 to 2006. Traces have been analyzed independently for each direction, so as to study asymmetry and differences between the US to Japan and Japan to US anomalies.

### 4.2 Analysis methodology

The parameters of the detection procedure are set to: $N = 8, M = 32, \Delta_0 = 5\text{ms}$, $\Delta_j = \Delta_0 2^j$, with $j = 1, ..., J$, and $J = 10$ ; $\lambda = 0.5$ (see also Table 1). Robustness of the results described below with respect to variations of these parameters has been checked. Let us note first that the choice of $N, M$ results from a trade-off: Increasing the number of sketch outputs $M$ reduces the number of sketches $N$ that are necessary for identification, hence diminishes the computational cost; However too large $M$ may result in too low aggregated volume of traffic per sketch, hence in a failure of the multiresolution gamma modeling. The choices of $\Delta_0$ and $J$ are motivated by previous work [1, 4, 16] showing that the relevant time scales for anomalies range from 1ms to 1s. The choice of the threshold $\lambda$ is motivated by previously obtained ROC (cf. [1, 4]).

The use of a detection tool on a database where anomalies are not labeled requires *a posteriori* validations. We have verified for all studied traces that, whenever a (collection of) attribute(s) is detected as associated to an anomaly, this corresponds to either of the two following situations. First, inspection of the packets sharing this attribute(s) reveal well-known anomalies or attacks (DDoS, flooding, portscan,...). Second, there

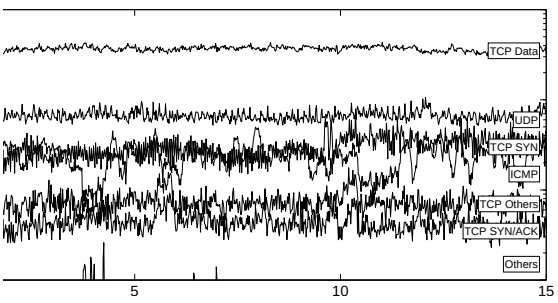


Figure 1: **Case study 1: Aggregated traffic** (Japan to USA) split by protocols, aggregation over 1s. Protocol split is here for convenience, but not actually used in the detection procedure.

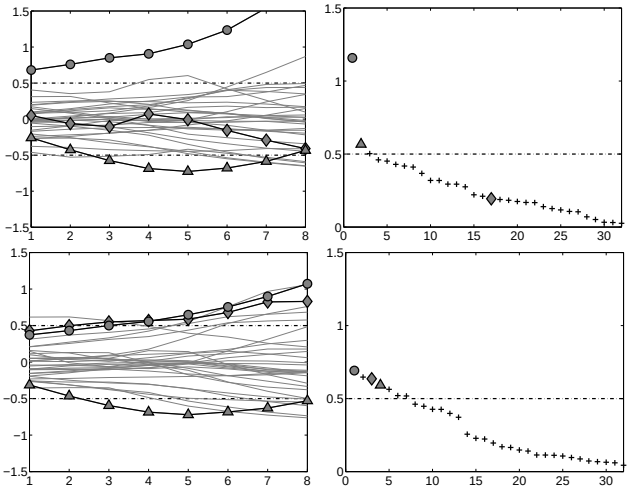

Figure 2: **Case study 1: Distances for two different hash functions.** Left: $(\alpha_{\Delta_j}^{n,m} - \alpha_{\Delta_j}^{m,R})/\sigma_{m,\alpha,\Delta_j}$ vs. $j$ curves; most curves, but two (top plot) or three (bottom plot), fall within the $\pm\lambda$ *normality* band for all $j$. Right: sorted $D_{\alpha^{n,m},m\in 1,\ldots,M}$; One sees that several sketch outputs have distances above $\lambda$. Combining sketches, one raises alarms by retaining only IPdst whose distances remain consistently above threshold for all hash functions. Here, circles, triangles and diamonds mark the sketch outputs containing the mixed flooding attack, the DNS anomaly and the SSH transfer, respectively.

are instances where the detection is associated to unexpected traffic features that correspond to an activity that cannot be identified (new protocol, dysfunctions, in some cases *elephants*, and maybe new attacks). In any case, traffic associated to alarms is not regular. On the other hand, inspection of the traces by an expert shows that when choosing at random sketch outputs with low distance (i.e., where no detection is made), no anomaly amongst flows carrying more than 1% of the total volume of the traffic can be identified. Hence, detection performance in terms of false positive and false negative is satisfactory. The remainder of this section details with the analysis of two representative case studies, and comment the general results about anomalies obtained from the database.

## 4.3 Case study 1: low-intensive long-lasting spoofed flooding

We first demonstrate the ability of detecting low-intensive long-lasting anomalies in a trace. Fig. 1 shows the aggregated time series of a 15-min sample trace, split by protocols to illustrate that no obvious anomaly can be seen (by eye), even with such a representation. Note however that detection is made without this split by protocol, IPdst is used as the hashing key. Fig. 2 shows distances for the $M = 32$ outputs for two given hash functions. One sketch output (left plots, circle) is constantly (for all $j$) above the detection threshold $\lambda$, systematically yielding the largest distance $D_{\alpha^{n,m}}$ (right plots) and hence an alarm. However, for a single hash function, several outputs are above the threshold (cf. Fig. 2, left plots), corresponding to many IPdst (23670 out of the 189361 in the trace). Table 2 demonstrates how, by combining up to 8 different hash functions so as to rule out random collisions, the detection procedure raises finally only one alarm, here associated to an anomaly on a single IPdst (marked with circles in Fig. 2). If one filters out anomaly amid the IPdst receiving at least 1000 packets, only $N = 6$ hash functions would have been necessary to identify this anomaly (cf. Table 2 last row). By packet inspection, the detected anomaly is identified as a mixed flooding attack against a single IPdst. Several protocols are used simultaneously: TCP (no SYN packets), UDP and ICMP. The distribution of packet size also follows a profile that mimics that of real traffic to make it harder to detect the attack. IP source addresses are spoofed and destination port is 0. These features clearly revealed the attack nature of this anomaly.

Fig. 2 indicates that, for a number of different hash functions, two other sketch outputs (marked with triangles and diamonds) often produce distances $D_{\alpha^{n,m}}$ above $\lambda$. Fig. 2 (left plots) shows that it is mostly due to distances being above $\lambda$ for specific ranges of time-scale $j$: around 10ms for triangles and 50ms and up for diamonds. Table 2 indicates that one of these alarms (triangles in Fig. 2) remains until the use of the 6-th hash function and is then removed by further sketching. Investigating traffic reaching the corresponding IPdst, we found that it corresponds to DNS traffic having a specific, periodic structure which appears as anomalous insofar as such a periodicity is not common for aggregated traffic. Also, we identify diamonds to be a long, but low volume, download via SSH protocol:

| N | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| $M_\lambda$ | | 4 | 2 | 5 | 7 | 4 | 3 | 3 | 4 |
| $\#\overline{IP}$ | 189361 | 23670 | 1479 | 231 | 50 | 6 | 0 | 0 | 0 |
| $\#\overline{IP_*}$ | 710 | 88 | 5 | 1 | 0 | 0 | 0 | 0 | 0 |
| $\#IP$ | 189361 | 23472 | 1444 | 249 | 56 | 11 | 2 | 2 | 1 |
| $\#IP_*$ | 710 | 90 | 6 | 2 | 2 | 2 | 1 | 1 | 1 |

Table 2: Case study 1: Identification. N: number of hash functions used. $M_\lambda$: number of sketch outputs above threshold $\lambda$; $\#\overline{IP}$: expected number of IPdst falling in those outputs; $\#\overline{IP_*}$: same as previous for IPdst receiving more than 1000 pkts; $\#IP$: actual number of IPdst belonging to those sketch outputs, hence raising alarm; $\#IP_*$: same as previous for IPdst receiving more than 1000 pkts. This shows that combining a reasonable number of hash functions, IPdst with anomalies are found, whereas the expected number of accidental collisions goes to zero.

For some hash functions, it is classified as anomalous because it also exhibits some form of periodicity, with a period larger than 100 ms. It might happen that, depending on the remainder of the traffic that fall within the same sketch output, this SSH download dominates at large scales and hence causes the distance to bypass the threshold. However, both DNS and SSH download consist of legitimate traffic. It is therefore a satisfactory output of our detection tool that the use of a large enough number of sketches removes them from the list of anomalies.

Let us further mention that the detected mixed flooding with spoofing attack corresponds to only 1% of the traffic and is by far not the largest elephant in the trace; this illustrates that our procedure is not simply focused on volume anomalies. Moreover, the anomaly lasts for the entire fifteen minutes of the analyzed trace. Because detection arises from comparisons between sketch outputs containing normal traffic and those carrying anomalous traffic, it does not require that the anomaly starts within the analyzed time window and does not rely on a change in time: hence the procedure is not fooled here. To validate that no anomaly was missed, we have systematically inspected traffic towards IPdst that received more than 0.1% of the total volume of the traffic and checked that no other anomaly could be identified.

## 4.4 Case study 2: short-lived portscan

Next, we focus on detection and near real-time tracking of short-lived portscan anomalies. Fig. 3 (top left) shows a directional aggregated time series. Detection is conducted using IPsrc as the hashing key. Fig. 3 (bottom left) shows distances that for a given hash function

| N | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| $M_\lambda$ | | 13 | 11 | 8 | 11 | 13 | 13 | 13 | 12 |
| $\#\overline{IP}$ | 35365 | 14367 | 4938 | 1234 | 424 | 212 | 86 | 34 | 12 |
| $\#\overline{IP_*}$ | 376 | 152 | 52 | 13 | 4 | 1 | 0 | 0 | 0 |
| $\#IP$ | 35365 | 14326 | 5031 | 1276 | 470 | 194 | 77 | 44 | 21 |
| $\#IP_*$ | 376 | 163 | 53 | 13 | 8 | 7 | 6 | 6 | 5 |

Table 3: Case study 2: Identification. Same legend as Table 2.

(#3 in Table 3): 8 sketch outputs are above threshold $\lambda$. Note that distances $D_{\alpha^{n,m}}$ (for $m \in 1, ..., M$) decrease more slowly than in Fig. 2, where IPdst is the hashing key. Combining the $N = 8$ hash functions finally retain 21 IPsrc addresses, amongst which only 5 emit more than 1000 packets (cf. Table 3), retained as meaningful anomalies.

An *a posteriori* inspection of the corresponding packets shows that 3 of them can be identified as port scanning. The first detected portscan aims at finding FTP servers, it corresponds to only 0.9% of the total volume of the traffic. The aggregated time series corresponding to the output of one hash function (#3) containing this anomaly is depicted in Fig. 3 (top right) and reveals that this anomaly lasts for a little more than 4 minutes. The second detected portscan consists of bursts of a few seconds gathering around 2% of the total volume of the traffic, and tracks HTTP servers. The third one searches for all open ports in hosts over a small subnetwork (around a few thousands hosts, 0.6% in volume, one minute long). The fourth anomaly consists of a HTTP flood, i.e., a large number of what can be seen as HTTP requests (4% of the traffic); all source ports are identical hence excluding simple parallel downloading. The fifth anomaly is a heavy traffic (4.5%) using GRE (generic routing encapsulation) protocol. GRE packets likely carry IPv6 traffic in the link, though it is difficult to decide whether it consists of legitimate or illegitimate traffic.

Let us further illustrate that the detection procedure is not simply volume based. It does not simply detect the largest *elephant* of the trace: Indeed, the smallest (in packet number) detected anomaly is only the 20th largest *elephant*. Also, we have checked that the 15 other largest *elephants* are not classified as anomalies and consist of normal traffic (HTTP exchanges, TCP packets towards non standard ports, IRC, other data that does not seem to resort to worm activity). Finally, the technique has the potential to be used over much shorter time windows. Fig. 3 (bottom right) depicts, for the first portscan anomaly, the distances $D_{\alpha^{n,m}}$ vs. time, computed using $T = 1$ min long sliding windows. For the entire duration of the portscan (corresponding to 6 time-windows), distances are above the threshold.

Hence that the procedure perfectly locates the anomaly in time, without the recourse of rupture or a volume based detection procedures. That is, the anomaly is seen because its short time correlation structures differ from those of the normal or background traffic.

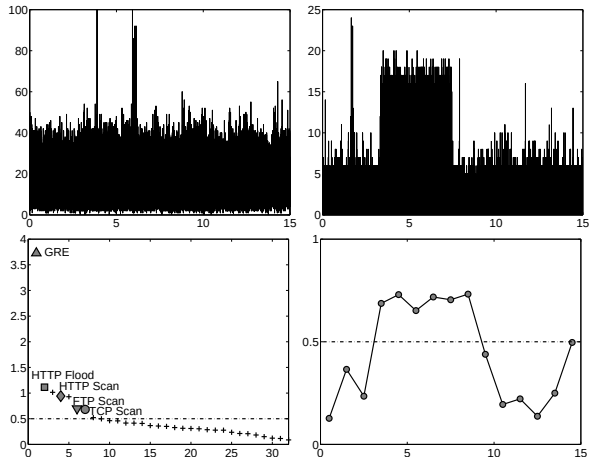


Figure 3: **Case study 2.** Top: aggregated time series of full traffic (left) and for a sketch output where a FTP portscan anomaly is detected (right), aggregation over 5ms. Bottom left: sorted $D_{\alpha^{n,m},m\in 1,\ldots,M}$; 7 sketch outputs are above the threshold $\lambda$. Bottom right: $D_{\alpha^{n,m},m\in 1,\ldots,M}$ computed within $T = 1$min time windows for the sketch output plotted above. The portscan lasts only a few minutes, between minutes 4 to 9 and is well detected at its precise times of occurrence.

## 4.5 Typology of the anomalies

Let us now turn to a higher level description of the anomalies found by means of our procedure. The most surprising result is the observation that anomalies were found in almost every trace over the 6 years and, in most cases, traces contained not a single but many anomalies. Also, the average number of detected anomalies is increasing over the years, and so is their variety. In 2001, anomalies mostly consisted of straightforward flooding attacks. Over the years, there is an increasing number of detected anomalies that turn out to be hard to identify. Some correspond, for instance, to IPdst receiving a small number of packets from a large number of different hosts (hundreds or thousands) on higher ports. This may be related to peer-to-peer (P2P) nodes/servers traffic. No further information is available to decide whether this is legitimate or not. *Elephants* corresponding to HTTP traffic (and in some cases to FTP or SSH connections) are sometimes detected as anomalies, but this tends to decrease over the years because of the increase of the number of used IP addresses found in the traces: hashing efficiency for obtaining normal traffic reference is improved by the multiplication of background traffic. Traffic is found asymmetric on this trans-Pacific link, and there are differences even in the most frequently observed anomalies depending on the direction of traffic. Notably, for large scale attacks, SYN flooding is commonly observed from US to Japan while ICMP flooding is more usual in the opposite direction.

When using IPdst as a hashing key, we found a variety of anomalies. There are a lot of flooding of various kinds (UDP, TCP-SYN, TCP, ICMP, sometimes mixed), attacking sources being either isolated, distributed (typically, a couple of dozen of sources) or spoofed (fixed "impossible" IP of the same subnetwork or thousands of random IPs). The involved ports are often known ones (FTP, SSH, HTTP, MySQL). Sometimes, they are selected randomly (anomalies towards a single high port or several different ports), and more rarely consist of invalid ports (port 0). Also, we found anomalies related to DNS traffic (very regular, for small periods of a few tens of seconds) from a limited number of hosts. Some anomalies consist of point-to-point GRE traffic. On very rare occasions (actually, a couple in 6 years), we observed systematic scans of all the ports of a single IP address.

When hashing on IPsrc, one naturally finds again some of the anomalies listed above. IPsrc hashing reveals a large variety of scan activities, mostly towards HTTP, SSH, MySQL, FTP. More recently, but still rarely, we found scans targeting the usual P2P ports or even larger sets of ports (some not identified). Some hosts responding to requests from a large number of different IP addresses are detected: this may be false alarms (HTTP, MySQL traffic) but it is associated from time to time to some traffic of worms/virus. Note that, by construction, the procedure does not aim at detecting worms/virus packets that are best found from their signature. However, it can detect the outburst of new worms when only a few hosts are infected (therefore having a traffic structure different from other, uncompromised hosts).

Concerning the duration of the anomalies, a large number of the detected ones (such as flooding/transfer anomalies) last for much longer than the entire 15 minutes (this does not fool our detection procedure which needs not observe beginnings or ends of the anomaly to be efficient). There are also small bursts, with duration ranging from a couple of seconds to a few minutes, especially of SYN and ICMP floodings. As previously mentioned, they can be precisely located using short (1min or 30sec) analysis time windows. We have found attacks lasting longer than the day, and one of them kept going in all traces over 9 months. Scans are usually much shorter, typically in a few minutes. Some of them consist of the regular repetition of a brief pattern.

Rare cases of large scale attacks (containing up to 30% of traffic) were also observed. However, most anomalies have much lower volume, and we tracked all anomalies with volumes down to 0.1% of the total traffic volume. They are hidden behind *elephants* of regular traffic in most cases, yet correctly detected.

# 5 Conclusions and Perspectives

The detection tool proposed here appears to be particularly relevant to extracting anomalies from single-point backbone measurements. We were able to find many anomalies in the database of the MAWI repository, in an unexpectedly large number. This is the first report on labeling the traces in the repository with anomalies, and further analysis on this database is to be pursued.

Based on its multiresolution properties, the detection tool is able to detect short-lived anomalies as well as longer ones; we have put the emphasis on the fact that the procedure should not be reduced to a rupture change, or a volume-based detection, as many profile-based are: due to the non Gaussian modeling used as its background, the detection method is sensitive to the statistical characteristics (short time correlations) of anomalies hidden in large scale traffic.

This detection tool benefits from a very low computational cost so that one can easily think of real-time (on-line, and on-the-fly) implementation and hence mitigation, even on loaded backbone networks. The choice of the type of attribute that is used as input for the hash function determines the type of anomalies that are investigated: For instance, in the present study, hashing the IPdst address is more intended toward the detection of flooding attacks, while selecting the IPsrc address is prone to reveal port scan operations. In particular, an improvement of the method would be hashing jointly with respects to two or more attributes, that could be fruitful to detect other kinds of anomalies. Still, the present study has covered almost all the well-known, classical anomalies that are usually found and enabled the automatic discovery of a number of new anomalies whose nature is being investigated.

One practical issue for further research is to add a feature to filter specific known anomalies. As in the DNS traffic in our results, legitimate traffic could be identified as an anomaly if it has a unique traffic pattern. Thus, to reduce false alarms, there must be a way to filter certain traffic patterns once they are labeled as legitimate by other means.

Another important issue is to evaluate the algorithm against sampled traffic or NetFlow data, which is particularly of interest to backbone network operation.

# References


[1] Abry, P., Borgnat, P., and Dewaele, G. Sketch based anomaly detection, identification and performance evaluation. In *IEEE/IPSJ SAINT Measurement Workshop* (Jan. 2007).

[2] Barford, P., Kline, J., Plonka, D., and Ron, A. A signal analysis of network traffic anomalies. In *IMW* (Nov. 2002).

[3] Basseville, M. Distance measures for signal processing and pattern recognition. *Signal Processing 18* (1989), 349–369.

[4] Borgnat, P., and et al. Détection d'attaques de dénis de service par un modèle non gaussien multirésolution. In *CFIP-2006* (Nov. 2006).

[5] Brutlag, J. Aberrant behavior detection in time series for network monitoring. In *USENIX System Administration Conference* (Dec. 2000).

[6] Cho, K., Mitsuya, K., and Kato, A. Traffic data repository at the WIDE project. In *USENIX FREENIX Track* (http://mawi.wide.ad.jp/mawi, June 2000).

[7] Huang, P., Feldmann, A., and Willinger, W. A non-intrusive, wavelet-based approach to detecting network performance problems. In *IMW* (Nov. 2001).

[8] Jin, S., and Yeung, D. A covariance analysis model for DDoS attack detection. In *ICC* (June 2004).

[9] Kim, Y., Lau, W. C., Chuah, M. C., and Chao, H. J. Packetscore: A statistics-based packet filtering scheme against distributed denial-of-service attacks. *IEEE Trans. Dependable Secur. Comput. 3*, 2 (2006), 141–155.

[10] Krishnamurty, B., Sen, S., Zhang, Y., and Chen, Y. Sketch-based change detection: Methods, evaluation, and applications. In *IMC* (Oct. 2003).

[11] Lakhina, A., Crovella, M., and Diot, C. Diagnosing network-wide traffic anomalies. In *SIGCOMM* (Aug. 2004).

[12] Li, X., Bian, F., Crovella, M., Diot, C., Govindan, R., Iannaccone, G., and Lakhina, A. Detection and identification of network anomalies using sketch subspaces. In *IMC* (Oct. 2006).

[13] Muthukrishnan, S. Data streams: Algorithms and applications. In *SODA* (Jan. 2003).

[14] Park, K., and Willinger, W., Eds. *Self-Similar Network Traffic and Performance Evaluation*. Wiley, 2000.

[15] Patwari, N., and Hero, A. Manifold learning visualization of network traffic data. In *MineNet* (Aug. 2005).

[16] Scherrer, A., Larrieu, N., Owezarski, P., Borgnat, P., and Abry, P. Non gaussian and long memory statistical characterisations for internet traffic with anomalies. *IEEE Trans. Dependable Secur. Comput.*, 1 (Jan. 2007), 56–70.

[17] Soule, A., Salamatian, K., and Taft, N. Combining filtering and statistical methods for anomaly detection. In *IMC* (Oct. 2005).

[18] Thorup, M., and Zhang, Y. Tabulation based 4-universal hashing with applications to second moment estimation. In *SODA* (Jan. 2004).

[19] Zhang, Y., Ge, Z., Greenberg, A., and Roughan, M. Network anomography. In *IMC* (Oct. 2005).